REMARKS

Claims 1-43 remain pending in the current Application. No amendments to the claims are being made herein.

Provisional Obviousness-type Double Patenting of Claim 1

Applicant respectfully submits that claims 1-43 are patentably distinct over Application No. 10/657,331. However, in order to further prosecution, Applicant is submitting concurrently herewith a terminal disclaimer to obviate the obviousness double-patenting rejection over Application No. 10/657,331. Applicant also respectfully submits that claims 1-43 are patentably distinct over Application No. 10.657,510. However, in order to further prosecution, Applicant is submitting concurrently herewith a terminal disclaimer to obviate the obviousness double-patenting rejection over Application No. 10/657,510. Therefore, Applicant respectfully requests that the Examiner withdraw the provisional rejection of obviousness-type double patenting.

Rejection of claims 1, 2, 6, 7, 10, 12, 16, 17, 19, 20, 21, 25, and 27 under 35 U.S.C. 102(b)

Applicants respectfully submit that claims 1, 2, 6, 7, 10, 12, 16, 17, 19, 20, 21, 25, and 27 are patentable over US Patent No. 4, 760, 545 (hereinafter referred to as Inagami) under 35 U.S.C. 102(b).

Claims 1 and 16

Applicants submit that claims 1 and 16 are not taught or suggested by Inagami, because Inagami does not teach or suggest each and every element of the claims. For example, each of claims 1 and 16 includes one or more instructions which specify a first offset between data elements within a first portion of successive data elements in the memory and a second offset between the first portion and a second portion of data elements in the memory. Applicants submit that Inagami at least does not teach or suggest the second offset as claimed. The Examiner indicates that D1 of Inagami teaches the second offset "between the first portion (the portion of elements being transferred started at D1) and a second portion [VAR] of data elements" where VAR is the top address in memory. However, D1 simply indicates a read/write start element number within a vector register (the vector register indicated by R1), but does not

indicate an offset between a first portion of data elements in the memory and VAR (see FIG. 4b and col. 4, lines 56-61, of Inagami). D1 provides an offset for a start element *from the start of a vector register* (by providing the start element number), but does not provide an offset between two portions of data elements in the memory, as claimed. Therefore, for at least these reasons, Applicants submit that claims 1 and 16 are not taught or suggested by Inagami.

The Examiner, in his response to the above argument (in paragraphs 18 and 19 of the current Office Action), states that "a starting location [D1] of the register is an offset between the first portion before the starting location and the second portion (after the starting location) of the data element in the memory." However, claims 1 and 16 claim that the one of the one or more instructions specifies "a first offset between data elements within a first portion of successive data elements in the memory", and the Examiner indicates that this is taught by VIR of Inagami, where VIR stores "the distance or displacement between the elements of the vector data *on the main memory*" (emphasis added, see col. 4, lines 62-65). Therefore, if the Examiner indicates that VIR teaches the first offset, then the "first portion" of data elements are data elements which reside in the main memory of Inagami and not within register R1. Therefore, it is improper for the Examiner to now state that the first portion "is before the starting location" of D1 in the register R1." That is, the Examiner cannot be inconsistent with the mapping of the claims elements. Furthermore, the Examiner holds that "registers are components of a memory in the hardware level... therefore, there is no distinction between memory and register unless applicant can show his memory is not using registers." However, throughout the specification of the current Application, general purpose registers and memory are used differently because they serve separate purposes. Furthermore, even Inagami makes a distinction between main memory and registers and thus the Examiner, in using Inagami, cannot ignore the distinction made in Inagami between registers and memory in the rejection of the claims. However, in order to further prosecution and not for prior art reasons, Applicants have amended claims 1 and 16 to further clarify that "the memory" of claims 1 and 16 does not include the "at least one general purpose register." That is, as seen clearly in the example illustration of FIG. 1 of the current Application, memory such as memory 12 can be separate and not include the general purpose registers, such as the general purpose registers in register file 34. Therefore, for at least these additional reasons, Applicants submit that claims 1 and 16 are allowable over Inagami.

Claims 2-14 and 17-30 have not been independently addressed because they depend directly or indirectly from allowable claim 1 or 16, and are therefore allowable for at least those reasons which apply to claims 1 and 16.

Rejection of claims 15, 30, 31, 32, 33, 35, 36, 38, 39, 41, and 42 under 35 U.S.C. 103

Applicants respectfully submit that claims 15, 30, 31, 32, 33, 35, 36, 38, 39, 41, and 42 are patentable over Inagami in view of US Patent No. 6,898,691 (hereinafter referred to as Blomgren) under 35 U.S.C. 103.

With respect to claims 15 and 31, ***one*** instruction of the one ore more instructions specifies a radix specifier. With respect to claim 32, ***one*** instruction of the one or more instructions implements storing predetermined data elements in the memory in a bit-reversed order and transfers the predetermined data elements into the at least one general purpose register. With respect to claim 35, ***one*** instruction of the one or more instruction transfers predetermined data elements into at least one general purpose register in a bit-reversed order. With respect to claim 38, ***one*** instruction of the one ore more instructions implements storing predetermine data elements in the at least one general purpose registers in a bit-reversed order and transfers the predetermined data elements into memory. With respect to claim 41, ***one*** instruction of the one ore mores instruction stores predetermined data elements in at least one register in sequential order and transfers the predetermined elements into the memory in a bit-reversed order.

The Examiner agrees that Inagami does not specifically show a radix specifier for implementing the data element transfer in a bit reverse order as claimed. However, the Examiner cites Blomgren as teaching a vector system including instruction for specifying a radix and transferring data element in reverse order. Applicants respectfully disagree. Firstly, the system in Blomgren does not teach or suggest any one instruction which specifies a radix specifier as claimed, for example, in claims 15 and 31. The only instructions discussed in Blomgren are the block4 and block4v instructions. However, neither of these instructions specify a radix specifier. They simply transfer elements with matrix registers to rearrange the matrix data. The Examiner cites col. 9, lines 27-59, and col. 11, lines 34-42, as teaching a radix reversal order. However, these cited sections simply demonstrate how the block and block4v instructions may be used to perform address bit-reversal in order to implement a Fast Fourier Transfer. That is, multiple instructions (as discussed in col. 10, lines 1-27), including the block4v instruction are used to implement the bit-reversal; however, there is no teaching or suggestion of using a radix specifier within an instruction itself.

Secondly, Blomgren does not teach or suggest an instruction for transferring data element in bit-reverse order, as claimed in various ways in claims 15, 31, 32, 35, 38, and 41. Each of these claims require a *single* instruction (i.e. ***one*** instruction) to implement a bit-reversed transfer *between memory and registers.* As discussed above, Blomgren only discusses the block4 and

block4v instruction, which may be used *in addition to other instructions*, to perform address bit-reversal *once the data elements are in the matrix registers* (see, e.g., col. 6, lines 15-18). However, neither of these instructions is a single instruction which transfers data elements in bit-reverse order, and furthermore, neither of these instructions transfers data elements in bit-reverse order between memory and the matrix registers.

Thirdly, the there is no motivation in Inagami or Blomgren to implement a single instruction which includes a radix specifier or transfers elements as claimed. For example, even the radix function cited by the Examiner in Blomgren does not appear within an instruction, and also provides no motivation to include it within an instruction. That is, there is no suggestion in any of the cited references to include a radix specifier in an instruction or to provide a single instruction which can implement bit-reverse transfers between memory and registers. Also, the Examiner is using inappropriate hindsight to combine the instructions of Inagami with the radix discussion of Blomgren, since neither of these references suggest the higher complexity instructions as claimed. The Examiner also states that the modification is obvious because both Blomgren and Inagami sought the possibility to avoid reading and writing from a memory by reducing the number of independent memory transfers. However, this does not provide motivation to combine. The cited sections of Blomgren, for example, discuss the use of block4 and block4v instructions to implement a radix functions *once the data is already in the registers*, but does not affect the transfer of data between memory and the registers. Furthermore, the Examiner suggests that implementing the reversed bit transfer of Blomgren in Inagami would increase the ability of Inagami for processing vector operation with a higher degree of complexity. However, there is no motivation in Inagami to create a more complex system, which would be required if the vector instructions were modified as suggested by the Examiner.

The Examiner states, in response to the first two arguments regarding Blomgren (given in paragraphs 20 and 21 of the current Office Action), that "the bit reversal was already a radix bit (see col. 11, lines 34-42)... therefore, Blomgren did teach the radix specifier within the single instruction itself." However, this is incorrect. In referring to col. 11, lines 34-42, Blomgren states that "a radix-4 bit reversal can be accomplished by multiple use of the block4 instruction." That is, multiple block4 instructions can be used to implement a radix-4 bit reversal function, but this does not teach or suggest a radix specifier or bit within any particular instruction. Blomgren only teaches how to use *multiple instructions* to implement *a radix function*. Furthermore, the Examiner, in paragraph 21, states that "applicant's claim (e.g. claim 15) recite one or more instructions, not single instruction...applicant is reminded that unclaimed features cannot be used to overcome the prior art." However, Applicants respectfully disagree. For example, referring to

claim 15, while claim 15 claims "processor circuitry for executing one or more instructions," the claim proceeds to state "wherein **one of** the one or more instructions specifies a radix specifier for implementing transferring one or more data elements in a bit-reversed order between the memory and the at least one general purpose register." Therefore, "**one** of the one or more instructions" clearly refers to a **single** instruction. Similar language of "**one of** the one or more instructions" is also found in each of independent claims 31, 32, 35, 38, and 41. (Note that Applicants have amended claim 31 to correct a typographical error and maintain better consistency, and not for prior art reasons.)

The Examiner, in response to the third argument regarding the lack of motivation (given in paragraph 22 of the current Office Action), refers Applicants to the responses given in paragraphs 20 and 21 which have been addressed by Applicants above. The Examiner also states that "transfer between memory and register had been known, such as the load." However, this does not provide sufficient motivation to modify Blomgren in a way to achieve the claims of the current Application. Firstly, the transfer between memory and register is only one aspect of the claims, and, as discussed above, the cited art still fails to teach other aspects of the claims. Secondly, the block4 and block 4v instructions of Blomgren cited by the Examiner do not even deal with the transfer of data between memory and registers, but reorders data that is already present in the registers, and as discussed above, there is no motivation to use the block4 and block 4v instructions to transfer data between registers and memory.

Therefore, for at least these reasons, Applicants submit that claims 15, 31, 32, 35, 38, and 41 are allowable over the cited references. Claims 17-30, 33, 34, 36, 37, 39, 40, 42, and 43 have not been independently addressed because they depend directly or indirectly from allowable claim 15, 31, 32, 35 38, or 41, and are therefore also allowable for at least those reasons provided with respect to claims 15, 31, 32, 35 38, and 41.

Conclusion

The Office Action contains numerous statements characterizing the claims, the Specification, and the prior art. Regardless of whether such statements are addressed by Applicants, Applicants refuse to subscribe to any of these statements, unless expressly indicated by Applicants.

Applicants respectfully solicit allowance of the pending claims. Contact me if there are any issues regarding this communication or the current Application.

Respectfully submitted,

By: ______________________

CHIU, JOANNA G.
Attorney of Record
Reg. No.: 43,629
Telephone: (512) 996-6839
Fax No.: (512) 996-6854

SEND CORRESPONDENCE TO:

Freescale Semiconductor, Inc.
Law Department

Customer Number: 23125